Gödel's Incompleteness

X. Zhao

北京大学
PEKING UNIVERSITY

Outline

Outline

A Wildly Known Popularization

Consider a recursively axiomatic theory T which describes a given domain of objects $\mathcal{M}$ in language $\mathscr{L}$ in a manner we hope is complete. Moreover, suppose that T is capable of talking in its language $\mathscr{L}$ about its own syntax and proofs from its axioms. Now consider the sentence φ: "I am unprovable in $T = \mathrm{Th}(\mathcal{M})$" where "I" refers to φ.

A Wildly Known Popularization

Consider a recursively axiomatic theory T which describes a given domain of objects $\mathcal{M}$ in language $\mathscr{L}$ in a manner we hope is complete. Moreover, suppose that T is capable of talking in its language $\mathscr{L}$ about its own syntax and proofs from its axioms. Now consider the sentence φ: "I am unprovable in $T = \mathrm{Th}(\mathcal{M})$" where "I" refers to φ.

$$
\begin{array}{llllll}
\mathcal{M} & \models & \varphi & \Rightarrow & \varphi \in T & T = \mathrm{Th}(\mathcal{M}) \\
 & & & \Rightarrow & \varphi \text{ is provable in } T & \text{a contradiction to } \varphi; \\
\mathcal{M} & \not\models & \varphi & \Rightarrow & \varphi \notin T & T = \mathrm{Th}(\mathcal{M}) \\
 & & & \Rightarrow & \varphi \text{ is unprovable in } T & \\
 & & & \Rightarrow & \varphi \text{ is true in } \mathcal{M} & \text{a contradiction.}
\end{array}
$$

A Wildly Known Popularization

Consider a recursively axiomatic theory T which describes a given domain of objects $\mathcal{M}$ in language $\mathscr{L}$ in a manner we hope is complete. Moreover, suppose that T is capable of talking in its language $\mathscr{L}$ about its own syntax and proofs from its axioms. Now consider the sentence φ: "I am unprovable in $T = \mathrm{Th}(\mathcal{M})$" where "I" refers to φ.

$$
\begin{array}{lllll}
\mathcal{M} \models \varphi & \Rightarrow & \varphi \in T & & T = \mathrm{Th}(\mathcal{M}) \\
 & \Rightarrow & \varphi \text{ is provable in } T & & \text{a contradiction to } \varphi; \\
\mathcal{M} \not\models \varphi & \Rightarrow & \varphi \notin T & & T = \mathrm{Th}(\mathcal{M}) \\
 & \Rightarrow & \varphi \text{ is unprovable in } T & & \\
 & \Rightarrow & \varphi \text{ is true in } \mathcal{M} & & \text{a contradiction.}
\end{array}
$$

Conclusion: Clearly it's $T = \mathrm{Th}(\mathcal{M})$ that leads to the contradiction and then $T \neq \mathrm{Th}(\mathcal{M})$. Hence our goal of exhaustively capturing all theorems valid in $\mathcal{M}$ by means of the axioms of T has not been achieved and is in fact not possible, as we will show.

The General Proof Idea

- Before arriving at the destination, we should climb over three mountains.
- Arithmetize provability as a (partially) recursive predicate P;
- Show that every predicate can be represented by some formula which follows that the predicate P can be represented by $\mathsf{beb}(y)$;
- Prove the fixed point lemma (here also needs the second conclusion);
- At last through the fixed point lemma and the second conclusion we would conclude Gödel's first incompleteness theorem.

The General Proof Idea

- Before arriving at the destination, we should climb over three mountains.
- Arithmetize provability as a (partially) recursive predicate P;
- Show that every predicate can be represented by some formula which follows that the predicate P can be represented by $\mathsf{beb}(y)$;
- Prove the fixed point lemma (here also needs the second conclusion);
- At last through the fixed point lemma and the second conclusion we would conclude Gödel's first incompleteness theorem.

The General Proof Idea

- Before arriving at the destination, we should climb over three mountains.
- Arithmetize provability as a (partially) recursive predicate P;
- Show that every predicate can be represented by some formula which follows that the predicate P can be represented by $\mathsf{beb}(y)$;
- Prove the fixed point lemma (here also needs the second conclusion);
- At last through the fixed point lemma and the second conclusion we would conclude Gödel's first incompleteness theorem.

The General Proof Idea

- Before arriving at the destination, we should climb over three mountains.
- Arithmetize provability as a (partially) recursive predicate P;
- Show that every predicate can be represented by some formula which follows that the predicate P can be represented by $\mathsf{beb}(y)$;
- Prove the fixed point lemma (here also needs the second conclusion);
- At last through the fixed point lemma and the second conclusion we would conclude Gödel's first incompleteness theorem.

The General Proof Idea

- Before arriving at the destination, we should climb over three mountains.
- Arithmetize provability as a (partially) recursive predicate P;
- Show that every predicate can be represented by some formula which follows that the predicate P can be represented by $\mathsf{beb}(y)$;

- Prove the fixed point lemma (here also needs the second conclusion);
- At last through the fixed point lemma and the second conclusion we would conclude Gödel's first incompleteness theorem.

Definition 1.1

Our language is $\mathscr{L}_A$ which is consist of non-logical symbols and logical symbols as follows:

- non-logical symbols: $\bar{0}$, $\bar{S}$, $\bar{+}$, $\bar{\times}$;
- logical symbols: $x_0, x_1, x_2, \cdots,\ \bar{=}, \neg, \rightarrow, \forall$.

Definition 1.2

For convenience, we define

$$\begin{array}{rcl} \sigma \bar{\neq} \tau & \text{iff} & \neg(\sigma \bar{=} \tau); \\ \bigvee_{i<n} \varphi_i & \text{iff} & \varphi_0 \vee \cdots \vee \varphi_{n-1}; \\ \bigwedge_{i<n} \varphi_i & \text{iff} & \varphi_0 \wedge \cdots \wedge \varphi_{n-1}. \end{array}$$

Definition 1.3

Φ is a set of the following logical axioms.

- (P_1) $\varphi \to (\psi \to \varphi)$;
- (P_2) $(\varphi \to \psi \to \vartheta) \to (\varphi \to \psi) \to (\varphi \to \vartheta)$;
- (P_3) $(\neg\varphi \to \psi) \to (\neg\varphi \to \neg\psi) \to \varphi$;
- (S) $\forall x\varphi \to \varphi(x;\tau)$ where $\varphi(x;\tau)$ is a free substitution;
- (D) $\forall x(\varphi \to \psi) \to \forall x\varphi \to \forall x\psi$;
- (E_1) $\tau \equiv \tau$;
- (E_3) $\tau_0 \equiv \sigma_0 \to \cdots \to \tau_{n-1} \equiv \sigma_{n-1} \to F(\tau_0, \cdots, \tau_{n-1}) \equiv F(\sigma_0, \cdots, \sigma_{n-1})$;
- (C_1) $\varphi \to \forall x\varphi$ where $x \notin \mathrm{Fr}(\varphi)$;
- (C_2) $\forall x_0 \cdots \forall x_{n-1}\varphi$ where φ is an axiom with one of the above forms.
- (MP) $\{\varphi, \varphi \to \psi\} \vdash \psi$.

Robinson Arithmetic

Definition 1.4

Robinson arithmetic is the theory $\mathsf{Q} = \Phi + \{Q_1, \cdots, Q_7\}$.

$$
\begin{aligned}
Q_1 &: \forall x\, \bar{S} x \mathrel{\bar{\neq}} \bar{0};\\
Q_2 &: \forall x \forall y (\bar{S} x \mathrel{\bar{\equiv}} \bar{S} y \to x \mathrel{\bar{\equiv}} y);\\
Q_3 &: \forall x (x \mathrel{\bar{\neq}} \bar{0} \to \exists y (x \mathrel{\bar{\equiv}} \bar{S} y));\\
Q_4 &: \forall x (x \mathbin{\bar{+}} \bar{0} \mathrel{\bar{\equiv}} x);\\
Q_5 &: \forall x \forall y (x \mathbin{\bar{+}} \bar{S} y \mathrel{\bar{\equiv}} \bar{S}(x \mathbin{\bar{+}} y));\\
Q_6 &: \forall x (x \mathbin{\bar{\times}} \bar{0} \mathrel{\bar{\equiv}} x);\\
Q_7 &: \forall x \forall y (x \mathbin{\bar{\times}} \bar{S} y \mathrel{\bar{\equiv}} x \mathbin{\bar{\times}} y \mathbin{\bar{+}} x).
\end{aligned}
$$

Attention: $\mathsf{Q} \vdash$ and $\vdash_{\mathsf{Q}}$.

Remark 1.5

1. $\mathcal{N}_0 = (\mathbb{N}, \mathtt{0}, S, +, \times)$ is a standard model of Q;
2. Also Q has many other models, for example, nonstandard models. $\mathcal{M} = (\mathbb{N} \cup \{\infty\}, \mathtt{0}, S, +, \times)$ where $S, +, \times$ are extensions to ∞ from $S, +, \times$ in $\mathcal{N}_0$ in the ways:
   - $S(\infty) = \infty$;
   - $n + \infty = \infty + n = \infty + \infty = \infty$;
   - $\mathtt{0} \times \infty = \infty \times \mathtt{0} = \mathtt{0}$ and $n \times \infty = \infty \times n = \infty \times \infty = \infty$.

   It's easy to check that $\mathcal{M} \models \mathsf{Q}$.

Robinson Arithmetic

Notation 1.6

The term number $\bar{n} = \bar{S}^n \bar{0} = \underbrace{\bar{S} \cdots \bar{S}}_{n \text{ many}} \bar{0}$ for all $n \in \mathbb{N}$.

Definition 1.7

$x \bar{\leq} y$ if $\exists z (z \bar{+} x \bar{=} y)$. Furthermore $x \bar{<} y$ iff $x \bar{\leq} y \wedge x \bar{\neq} y$.

Robinson Arithmetic

Lemma 1.8

1. $\mathsf{Q} \vdash \forall x(\bar{S}x \bar{+} \bar{n} \bar{=} x \bar{+} \bar{S}\bar{n});$
2. $\mathsf{Q} \vdash \overline{m} \bar{+} \bar{n} \bar{=} \overline{m+n};$
3. $\mathsf{Q} \vdash \overline{m} \bar{\times} \bar{n} \bar{=} \overline{m \times n};$
4. $\mathsf{Q} \vdash \overline{m} \bar{=} \bar{n}$ *iff* $m = n;$
5. $\mathsf{Q} \vdash \overline{m} \bar{\leq} \bar{n}$ *iff* $m \leq n;$
6. $\mathsf{Q} \vdash \forall x(x \bar{\leq} \bar{n} \leftrightarrow \bigvee_{q \leq n} x \bar{=} \bar{q});$
7. $\mathsf{Q} \vdash \forall x(x \bar{\leq} \bar{n} \vee \bar{n} \bar{\leq} x).$

Computability

Definition 1.9

Suppose $f : \mathbb{N}^k \to \mathbb{N}$ is a function, we say f is computable if, there is some algorithm set in some fixing machine such that, given any $x_0, \cdots, x_{n-1}$, when inputting $x_0, \cdots, x_{n-1}$ into the machine,

- the machine would output $f(x_0, \cdots, x_{n-1})$ if $f(x_0, \cdots, x_{n-1})$ is defined;
- the machine would never stop if $f(x_0, \cdots, x_{n-1})$ isn't defined.

Computability

Lemma 1.10

The following basic functions are computable.

1. *the zero function* $0(x) = 0$ *for all* x*;*
2. *the successor function* $S(x) = x + 1$*;*
3. *the projection function* $U_i^n(x_0, \cdots, x_{n-1}) = x_i$ *for all* $i < n$.

Theorem 1.11 (Composition)

Suppose that $f(y_0, \cdots, y_{n-1})$ *and* $g_0(\overrightarrow{x}), \cdots, g_{n-1}(\overrightarrow{x})$ *are computable functions, then* $h(\overrightarrow{x}) = f(g_0(\overrightarrow{x}), \cdots, g_{n-1}(\overrightarrow{x}))$ *is also computable.*

Computability

Theorem 1.12 (Recursion)

Suppose that $f(\overrightarrow{x})$ and $g(\overrightarrow{x}, y, z)$ are functions, define $h(\overrightarrow{x}, y)$ by recursion equations: $h(\overrightarrow{x}, 0) = f(\overrightarrow{x})$ and $h(\overrightarrow{x}, y+1) = g(\overrightarrow{x}, y, h(\overrightarrow{x}, y))$. If $f(\overrightarrow{x})$ and $g(\overrightarrow{x}, y, z)$ are computable, then so is $h(\overrightarrow{x}, y)$. And we say h is obtained by recursion from f and g.

Theorem 1.13 (Bounded Minimalisation)

Suppose $f(\overrightarrow{x}, y)$ is computable, then so is the following:

$$\mu z < y(f(\overrightarrow{x}, z) = 0) = \begin{cases} \text{the least } z < y & \text{if } \exists z f(\overrightarrow{x}, z) = 0; \\ y & \text{otherwise.} \end{cases}$$

And we call $\mu z < y$ as bounded minimalisation operator.

Theorem 1.14 (Minimalisation)

Suppose that $f(\overrightarrow{x}, y)$ is computable, then so is

$$\mu y(f(\overrightarrow{x}, y) = 0) = \begin{cases} \text{the least } y \text{ such that } f(\overrightarrow{x}, z) \text{ is defined} & \\ \text{for all } z < y \text{ and } f(\overrightarrow{x}, y) = 0 & \text{if there is such } y; \\ \text{undefined} & \text{otherwise .} \end{cases}$$

And we call μy as μ-operator.

Definition 1.15

Primitive recursive functions is the least class which includes identity maps ($n(x) = n$), projection functions and is closed under composition and recursion; Recursive functions is the least class which includes the basic functions and is closed under composition, recursion and μ-operator.

Definition 1.16

Suppose $P(\overrightarrow{x})$ is an n-ary predicate with $P \subseteq \mathbb{N}^n$. Define c_P as:

$$c_P(\overrightarrow{x}) = \begin{cases} 1 & P(\overrightarrow{x}) \text{ holds;} \\ 0 & \text{otherwise.} \end{cases}$$

$P(\overrightarrow{x})$ is recursive if c_P is computable; otherwise it's not recursive.

Lemma 1.17

Suppose that $P(\overrightarrow{x})$ *and* $Q(\overrightarrow{x})$ *are recursive predicates, then so are:*

1. *not* $P(\overrightarrow{x})$;
2. $P(\overrightarrow{x})$ *and* $Q(\overrightarrow{x})$;
3. $P(\overrightarrow{x})$ *or* $Q(\overrightarrow{x})$.

Lemma 1.18

Suppose that $P(\overrightarrow{x}, y)$ *is a recursive predicate, then so are:*

1. $Q_1(\overrightarrow{x}, y) = \forall z < y P(\overrightarrow{x}, z)$;
2. $Q_2(\overrightarrow{x}, y) = \exists z < y P(\overrightarrow{x}, z)$.

Definition 1.19

Suppose $A \subseteq \mathbb{N}$, if the characteristic function c_A of A given by

$$c_A(x) = \begin{cases} 1 & x \in A; \\ 0 & x \notin A. \end{cases}$$

is computable, we say A is recursive. And if the function f given by

$$f(x) = \begin{cases} 1 & x \in A; \\ \text{undefined} & x \notin A. \end{cases}$$

is computable, we say A is recursively enumerable.

Arithmetization

In $\mathscr{L}_A$ we can also talk about the syntax of $\mathscr{L}_A$, proofs, provability and even some semantics by means of encoding of strings from alphabet of $\mathscr{L}_A$, which is called Gödel encoding.

Definition 1.20

Assign to every symbol of $\mathscr{L}_A$ a natural number.

ζ	$\forall$	$\bar{0}$	$\bar{S}$	$\bar{+}$	$\bar{\times}$	(	)	$\neg$	$\to$	$\bar{=}$	x_0	x_1	x_2	$\cdots$
$\sharp\zeta$	1	3	5	7	9	11	13	15	17	19	21	23	25	$\cdots$

Then the Gödel code of a string $\xi = \zeta_0 \cdots \zeta_n$ is

$$\sharp\xi = \langle \sharp\zeta_0, \cdots, \sharp\zeta_n \rangle = p_0{}^{\sharp\zeta_0} \cdots p_n{}^{\sharp\zeta_n}.$$

In particular we set $\sharp\langle\ \rangle = 1$.

The sentence

$$\forall x(\bar{1} \overline{\neq} (\bar{2} \mathbin{\overline{\times}} x \mathbin{\overline{+}} \overline{21}))$$

not only states some assertion about 1 and 21, but also states some syntax of $\mathscr{L}_A$, i.e., "$\forall$ is not a variable".

Arithmetization

The sentence

$$\forall x(\,\bar{1}\ \bar{\neq}\,(\,\bar{2}\ \bar{\times}\ x\,\bar{+}\ \overline{21}\,))$$

not only states some assertion about 1 and 21, but also states some syntax of $\mathscr{L}_A$, i.e., "$\forall$ is not a variable".

All the syntactic concepts in which we are interested are "translated" as recursive subsets of $\mathbb{N}$, or at least recursively enumerable subsets of $\mathbb{N}$, to represent them in the theory $T \supseteq \mathsf{Q}$. Notably that we use $\sharp \cdots$ to denote the code of some $\cdots$.

The sentence

$$\forall x(\bar{1} \bar{\neq} (\bar{2} \bar{\times} x \bar{+} \overline{21}))$$

not only states some assertion about 1 and 21, but also states some syntax of $\mathscr{L}_A$, i.e., "$\forall$ is not a variable".

All the syntactic concepts in which we are interested are "translated" as recursive subsets of $\mathbb{N}$, or at least recursively enumerable subsets of $\mathbb{N}$, to represent them in the theory $T \supseteq \mathsf{Q}$. Notably that we use $\sharp \cdots$ to denote the code of some $\cdots$.

Lemma 1.21

$\{v \in \mathbb{N} | v \text{ is a } \sharp\text{variable}\}$ *is recursive.*

Proof.

Set $P = \{v \in \mathbb{N} | v = 2k + 21 \text{ for some } k \in \mathbb{N}\}$. Since the predicate $R(v, k) : v = 2k + 21$ is recursive, then the predicate $P(v) : \exists k < vR(v, k)$ is recursive. ○

Arithmetization

Lemma 1.22

$\{t \in \mathbb{N} | t \text{ is a } \sharp\text{term}\}$ *is recursive.*

Proof.

We just need to give a recursive definition of t:

- $\exists s < t(t = \langle s \rangle)$ where s is some $\sharp$varaible or s is $\sharp\bar{0}$;
- $\exists r, s < t(t = \langle r \rangle \hat{\ } s)$ where r is $\sharp\bar{S}$ and s is some $\sharp$term
- $\exists q, r, s < t(t = \langle q \rangle \hat{\ } s \hat{\ } t)$ where q is $\sharp\bar{+}$ or $\sharp\bar{\times}$ and s, t are $\sharp$terms.

In Lemma 1.22, given some term $\bar{S}\,\bar{S}\,\bar{0}$ the Gödel code of it is

$$\langle \sharp\bar{S}, \sharp\bar{S}, \sharp\bar{0} \rangle = 2^5 3^5 5^3$$

but

$$\langle \sharp\bar{S}, \langle \sharp\bar{S}, \sharp\bar{0} \rangle \rangle = 2^6 3^{\langle \sharp\bar{S}, \sharp\bar{0} \rangle} = 2^5 3^{2^5 3^3}.$$

Arithmetization

Lemma 1.23

$\{\varphi \in \mathbb{N} | \varphi \text{ is a } \sharp\text{(atomic formula)}\}$ *is recursive.*

Lemma 1.24

$\{\varphi \in \mathbb{N} | \varphi \text{ is a } \sharp\text{formula}\}$ *is recursive.*

Lemma 1.25

There is a recursive function sub *such that, for any term or formula* φ *and for any variable* x *and any term* t, $\mathrm{sub}(\sharp\varphi, \sharp x, \sharp t) = \sharp\varphi(x; t)$.

Arithmetization

Proof.

We also just give a recursive definition of sub.

$$\sharp\varphi(x;t) = \begin{cases} \sharp t & \varphi = x; \\ \sharp(\bar{S}(u(x;t))) & \varphi = \bar{S}u \text{ and } u \text{ is a term;} \\ \sharp(\bar{+}(u(x;t)s(x;t))) & \varphi = \bar{+}us \text{ and } u, s \text{ are terms;} \\ \sharp(\bar{\times}(u(x;t)s(x;t))) & \varphi = \bar{\times}us \text{ and } u, s \text{ are terms;} \\ \sharp(\bar{=}(u(x;t)s(x;t))) & \varphi = \bar{=}us \text{ and } u, s \text{ are terms;} \\ \sharp(\neg(\psi(x;t))) & \varphi = \neg\psi \text{ and } \psi \text{ is a formula;} \\ \sharp(\rightarrow \psi(x;t)\vartheta(x;t)) & \varphi = \rightarrow \psi\vartheta \text{ and } \psi, \vartheta \text{ are formulas;} \\ \sharp(\forall y(\psi(x;t))) & \varphi = \forall y\psi,\ y \neq x \text{ and } \psi \text{ is a formula;} \\ \sharp\varphi & \text{otherwise.} \end{cases}$$

This completes the proof. ◯

Arithmetization

Lemma 1.26

Define the predicate $P \subseteq \mathbb{N}^2$ *as*

$$\{(x, \varphi) \in \mathbb{N}^2 | x \text{ is a } \sharp(\text{free occurrence}) \text{ in } \varphi\}.$$

Then P *is recursive.*

Proof.

$(x, \varphi) \in P$ iff x is a $\sharp$variable $\wedge$ φ is a $\sharp$term or $\sharp$formula $\wedge$ $\text{sub}(\varphi, x, \sharp\bar{0}) \neq \varphi$. ∎

Lemma 1.27

$\{\sigma \in \mathbb{N} | \sigma \text{ is a } \sharp\text{sentence}\}$ *is recursive.*

Lemma 1.28

$\{\varphi \in \mathbb{N} | \varphi \text{ is a } \langle \sharp\neg \rangle\hat{\ }\langle \psi \rangle \wedge \psi \text{ is a } \sharp\text{formula}\}$ *is recursive.*

Lemma 1.29

$\{\varphi \in \mathbb{N} | \varphi \text{ is a } \langle \sharp \rightarrow \rangle\hat{\ }\langle \psi \rangle\hat{\ }\langle \vartheta \rangle \wedge \psi, \vartheta \text{ are } \sharp\text{formulas}\}$ *is recursive.*

Lemma 1.30

$\{\varphi \in \mathbb{N} | \varphi \text{ is a } \langle \sharp\forall \rangle\hat{\ }\langle y \rangle\hat{\ }\langle \psi \rangle \wedge y \text{ is a } \sharp\text{variable} \wedge \psi \text{ is a } \sharp\text{formula}\}$ *is recursive.*

Arithmetization

Lemma 1.31

Define the predicate $P \subseteq \mathbb{N}^3$ as

$\{(\varphi, x, t) \in \mathbb{N}^3 | \varphi$ is a $\sharp$formula $\wedge t$ is a $\sharp$term$\wedge x$ is a $\sharp$variable $\wedge\, t$ is a $\sharp$free for x in $\varphi\}$.

Then P is recursive.

Proof.

We give a recursive definition of "t is a $\sharp$free for x in φ":

- t is a $\sharp$free for x in φ, where φ is a $\sharp$(atomic formula);
- t is a $\sharp$free for x in ψ, where $\varphi = \langle \sharp\neg \rangle\hat{}\langle \psi \rangle \wedge \psi$ is a $\sharp$formula;
- t is a $\sharp$free for x in ψ and t is a $\sharp$free for x in ϑ, where $\varphi = \langle \sharp \to \rangle\hat{}\langle \psi \rangle\hat{}\langle \vartheta \rangle \wedge \psi, \vartheta$ are $\sharp$formulas;
- either x is not a $\sharp$(free ocurrence) in φ, or y is not a $\sharp$(free ocurrence) in t and t is a $\sharp$free for x in ψ, where $\varphi = \langle \sharp\forall \rangle\hat{}\langle \sharp y \rangle\hat{}\langle \psi \rangle \wedge y$ is a $\sharp$variable $\wedge\, \psi$ is a $\sharp$formula. ↻

Lemma 1.32

Define the predicate $P \subseteq \mathbb{N}^2$ *as*

$\{(\varphi, \psi) \in \mathbb{N}^2 | \varphi \text{ is a } \sharp(\forall\text{−comprehension}) \text{ of } \psi \text{ and } \varphi, \psi \text{ are } \sharp \text{ formulas}\}.$

Then P *is recursive.*

Proof.

$(\varphi, \psi) \in P$ iff $\exists x_0 < \varphi \cdots \exists x_{n-1} < \varphi(\varphi = \langle 1, x_0, \cdots, 1, x_n \rangle \hat{}\, \psi \wedge x_0, \cdots, x_{n-1}$ are $\sharp$variables $\wedge\, \psi$ is a $\sharp$formula).

Lemma 1.33

$\{\alpha \in \mathbb{N} | \alpha \text{ is a } \sharp\text{axiom}\}$ *is recursive.*

Definition 1.34

Suppose Γ is a set of formulas and T is a theory.

- Γ is recursive if $\sharp\Gamma = \{\sharp\varphi | \varphi \in \Gamma\}$ is recursive; otherwise we say Γ is not recursive;
- T is decidable if T is recursive; and T is undecidable otherwise.
- T is recursively axiomatizable if there is a recursive set Σ such that $T = T_\Sigma$, and we may say T is recursively axiomatized by Σ.

Arithmetization

Lemma 1.35

Let T be a theory and be recursively axiomatized by $X \subseteq T$, and define the predicates $\mathrm{Be}_T \subseteq \mathbb{N}^2$ and $\mathrm{Beb}_T \subseteq \mathbb{N}$ as

$$\{(p, \varphi) \in \mathbb{N}^2 | p \text{ is a } \sharp\text{proof of } \varphi \text{ in } T\} \textit{ and } \{\varphi \in \mathbb{N} | \exists x \, \mathrm{Be}_T(x, \varphi)\}$$

respectively, then Be_T is recursive and Beb_T is recursively enumerable.

Proof.

(1) Since $(p, \varphi) \in \mathrm{Be}$ iff

$$p \neq 1 \wedge (p)_{\mathrm{Length}(p)-1} = \varphi \wedge \forall k < \mathrm{Length}(p)$$
$$[(p)_k \in \sharp X \vee (p)_k \text{ is a } \sharp\text{axiom } \vee \exists i, j < k((p)_i = \langle \sharp \rightarrow \rangle \hat{\ } (p)_j \hat{\ } (p)_k)],$$

then Be_T is recursive;

(2) It's trivial that Beb_T is recursively enumerable. ↻

Definition 1.36

Fix our arithmetic language $\mathscr{L}_A$. The formulas $\varphi \in \Delta$ are defined recursively as follows:

- all the atomic formulas such as $\tau \equiv \sigma$, where τ, σ are terms, belong to Δ;
- if $\varphi, \psi \in \Delta$, then so $\neg\varphi, \varphi \to \psi \in \Delta$;
- if τ is a term with $x \notin \mathrm{Vr}(\tau)$, and $\varphi \in \Delta$, then so $\forall x \leq \tau \varphi \in \Delta$.

For any formula φ, $\varphi \in \Delta_0$ iff there is some $\psi \in \Delta$ ψ such φ and ψ are logically equivalent.

For any $\varphi \in \Delta_0$, $\exists \vec{x} \varphi \in \Sigma_1$ and $\forall \vec{x} \varphi \in \Pi_1$;

We say $\varphi \in \Delta_1$ if there is some $\psi \in \Sigma_1$ and $\vartheta \in \Pi_1$ such that φ, ψ, ϑ are logically equivalent.

Theorem 1.37 (Σ_1-completeness of Q)

For any Σ_1-sentence φ for $\mathscr{L}_A$, we have $\mathcal{N} \models \varphi$ iff $\mathsf{Q} \vdash \varphi$.

Definition 1.38

We say a k-ary predicate $P \subseteq \mathbb{N}^k$ is numeralwise representable or representable in T if, there is a formula $\varphi(\overrightarrow{x})$ for $\mathscr{L}_A$ such that for any $n_0, \cdots, n_{k-1} \in \mathbb{N}$,

$$(n_0, \cdots, n_{k-1}) \in P \Rightarrow T \vdash \varphi(\overline{n_0}, \cdots, \overline{n_{k-1}}),$$
$$(n_0, \cdots, n_{k-1}) \notin P \Rightarrow T \vdash \neg\varphi(\overline{n_0}, \cdots, \overline{n_{k-1}}).$$

We say a predicate $P \subseteq \mathbb{N}^k$ is Δ_0, or Σ_1, or Π_1 if it's represented by a Δ_0 formula, or Σ_1 formula, or Π_1 formula respectively. And if P can be represented by a Σ_1 formula and also be represented by a Π_1 formula, we say it's Δ_1.

Definition 1.39

Given any $\mathscr{L}_A$ formula $\varphi(\overrightarrow{x})$ and predicate $P \subseteq \mathbb{N}^k$, we say P is defined by $\varphi(\overrightarrow{x})$ in $\mathcal{M}$ iff for any $n_0, \cdots, n_{k-1} \in \mathbb{N}$ we have

$$(n_0, \cdots, n_{k-1}) \in P \Leftrightarrow \mathcal{M} \models \varphi(\bar{n}, \cdots, \overline{n_{k-1}}).$$

And if there is such φ we say P is definable in $\mathcal{M}$.

Some simple facts:

- Suppose T is a recursively axiomatizable theory. If P is representable, then P is recursive;
- It's easy to check that the class of representable predicates is closed under Boolean operators;
- P is representable in $\mathrm{Th}(\mathcal{N})$ iff P is definable in $\mathcal{N}$.

Definition 1.40

We say the function $f : \mathbb{N}^k \to \mathbb{N}$ is representable in $T \supseteq \mathsf{Q}$ if, there is a formula $\varphi(x_0, \cdots, x_{k-1}, y)$ such that, for all $n_0, \cdots, n_{k-1} \in \mathbb{N}^k$, we have

$$T \vdash \forall y[\varphi(\overline{n_0}, \cdots, \overline{n_{k-1}}, y) \leftrightarrow y \equiv \overline{f(n_0, \cdots, n_{k-1})}].$$

Similarly we say a function f is Δ_0, or Σ_1, or Π_1 if it's represented by a Δ_0, or Σ_1, or Π_1 formula respectively. And if f can be represented by a Σ_1 formula and also be represented by a Π_1 formula, we say it's Δ_1.

Suppose f is a function and $G_f = \{(x, y) | y = f(x)\}$.

- If φ represents f, then φ represents G_f;

Suppose f is a function and $G_f = \{(x,y) | y = f(x)\}$.

- If φ represents f, then φ represents G_f;
- φ represents G_f but φ may don't represent f. Set $Z(x) = \mathtt{0}(x) = \mathtt{0}$ and $G_Z = \{(x, \mathtt{0}) | x \in \mathbb{N}\}$. It's easy to check that the formula $y \bar{+} y \bar{\equiv} y$ represents the predicate G_Z. But since $\mathsf{Q} \nvdash \forall y(y \bar{\neq} \mathtt{0} \to y \bar{+} y \bar{\neq} y)$ (see Remark 1.5 (2)), then $y \bar{+} y \bar{\equiv} y$ doesn't represent the function $Z(x)$.

Suppose f is a function and $G_f = \{(x, y) | y = f(x)\}$.

- If φ represents f, then φ represents G_f;
- φ represents G_f but φ may don't represent f. Set $Z(x) = \mathtt{0}(x) = \mathtt{0}$ and $G_Z = \{(x, \mathtt{0}) | x \in \mathbb{N}\}$. It's easy to check that the formula $y \mathbin{\bar{+}} y \mathbin{\bar{\equiv}} y$ represents the predicate G_Z. But since $\mathsf{Q} \nvdash \forall y(y \mathbin{\bar{\not\equiv}} \mathtt{0} \to y \mathbin{\bar{+}} y \mathbin{\bar{\not\equiv}} y)$ (see Remark 1.5 (2)), then $y \mathbin{\bar{+}} y \mathbin{\bar{\equiv}} y$ doesn't represent the function $Z(x)$.
- f is representable iff G_f is representable.

Lemma 1.41

Let $\tau(x_0, \cdots, x_{k-1})$ be a term for $\mathscr{L}_A$, and define a function $f_\tau(n_0, \cdots n_{k-1}) = \tau(\overline{n_0}, \cdots, \overline{n_{k-1}})^{\mathcal{N}}$, and suppose $\mathsf{Q} \subseteq T$. Then f_τ is represented by $y \equiv \tau(x_0, \cdots, x_{k-1})$ in T. In particular, the zero function, the successor function, the projection functions, the constant functions, the plus function and the multiplication function are all representable.

Proof.

Since by induction on τ we can prove that for all $n_0, \cdots n_{k-1} \in \mathbb{N}$ $T \vdash \tau(\overline{n_0}, \cdots, \overline{n_{k-1}}) \equiv \overline{f_\tau(n_0, \cdots, n_{k-1})}$, then we have

$$T \vdash \forall y[y \equiv \tau(\overline{n_0}, \cdots, \overline{n_{k-1}}) \leftrightarrow y \equiv \overline{f_\tau(n_0, \cdots, n_{k-1})}]$$

for all $n_0, \cdots n_{k-1} \in \mathbb{N}$. So $y \equiv \tau(x_0, \cdots, x_{k-1})$ represents f_τ in T. ○

Lemma 1.42

Suppose $\mathsf{Q} \subseteq T$, *if* $h_0(\overrightarrow{x}), \cdots, h_r(\overrightarrow{x})$ *and* $\mathrm{g}(y_0, \cdots, y_{r-1})$ *are all representable in* T, *then so is* $f = \mathrm{g}(h_0, \cdots, h_{r-1})$.

Corollary 1.43

Suppose $\mathrm{g}(\overrightarrow{x}, y)$ *is representable in* $T \supseteq \mathsf{Q}$ *and* $\forall \overrightarrow{x} \exists y(\mathrm{g}(\overrightarrow{x}, y) = \mathsf{0})$, *then the function* $f(\overrightarrow{x}) = \mu y(\mathrm{g}(\overrightarrow{x}, y) = \mathsf{0})$ *is also representable.*

Suppose $f(\overrightarrow{x}, y)$ is defined by recursion with $g(\overrightarrow{x})$ and $h(x, y, z)$, that is: $f(\overrightarrow{x}, 0) = g(\overrightarrow{x})$ and $f(\overrightarrow{x}, n+1) = h(\overrightarrow{x}, n, f(\overrightarrow{x}, n))$.

Suppose $f(\overrightarrow{x}, y)$ is defined by recursion with $g(\overrightarrow{x})$ and $h(x, y, z)$, that is: $f(\overrightarrow{x}, 0) = g(\overrightarrow{x})$ and $f(\overrightarrow{x}, n+1) = h(\overrightarrow{x}, n, f(\overrightarrow{x}, n))$.

Recall that how we state the justice of recursions in set theory. We may give the explicit definition of $f(\overrightarrow{x}, n) = m$ by: there is an encoding number t of a finite sequence with length $n+1$ such that, $(t)_0 = g(\overrightarrow{x})$ and for all $i < n$, we have $(t)_{i+1} = h(\overrightarrow{x}, i, (t)_i)$ and $(t)_n = m$.

Suppose $f(\overrightarrow{x}, y)$ is defined by primitive recursion with $g(\overrightarrow{x})$ and $h(x, y, z)$, that is: $f(\overrightarrow{x}, 0) = g(\overrightarrow{x})$ and $f(\overrightarrow{x}, n+1) = h(\overrightarrow{x}, n, f(\overrightarrow{x}, n))$.

Recall that how we state the justice of recursions in set theory. We may give the explicit definition of $f(\overrightarrow{x}, n) = m$ by: there is an encoding number t of a finite sequence with length $n + 1$ such that, $(t)_0 = g(\overrightarrow{x})$ and for all $i < n$, we have $(t)_{i+1} = h(\overrightarrow{x}, i, (t)_i)$ and $(t)_n = m$.

In this process, we usually encode with the functions x^y and p_n. But the difficulty in showing the representability of them is the same as recursions. "Phoned with God", Gödel solved such difficulty with the help of Chinese reminder theorem, and as we seen in his method of encoding the finite sequences he used $+$ and $\times$ only instead of x^y and p_n.

Lemma 1.44

Suppose $\mathrm{g}(\overrightarrow{x})$ *and* $h(\overrightarrow{x}, y, z)$ *are representable in* $T \supseteq \mathsf{Q}$ *and* f *is defined by recursion with* g *and* h*, then* f *is representable in* T.

Theorem 1.45 (Representability)

For any recursive function f, f is representable in $T \supseteq \mathsf{Q}$ and Δ_1. Consequently every recursive predicate is representable in $T \supseteq \mathsf{Q}$ and Δ_1.

Corollary 1.46

For any predicate $P \subseteq \mathbb{N}^k$ and any recursively axiomatizable and consistent theory $T \supseteq \mathsf{Q}$, the following are equivalent.

1. *P is recursive;*
2. *P is representable;*
3. *P is representable and Δ_1.*

Notation 1.47

For any formula φ we use $\ulcorner \varphi \urcorner$ to denote the term $\bar{S}^{\sharp\varphi}\,\bar{0}$, i.e.,

$$\ulcorner \varphi \urcorner = \overline{\sharp\varphi} = \bar{S}^{\sharp\varphi}\,\bar{0}.$$

Lemma 1.48 (Fixed Point)

Given any $\mathcal{L}_A$ formula $\varphi(x)$ with only x free and a theory $T \supseteq \mathsf{Q}$, we can effectively find a sentence σ such that $T \vdash \sigma \leftrightarrow \varphi(\ulcorner \sigma \urcorner)$.

Definition 1.49

Let T be a theory for $\mathscr{L}_A$.

- We say T is ω-inconsistent if, there is an $\mathscr{L}_A$ formula $\varphi(x)$ such that $T \vdash \exists x\varphi(x)$ and $T \vdash \neg\varphi(\bar{n})$ for all $n \in \mathbb{N}$;
- We say T is ω-consistent if T is not ω-inconsistent, i.e., for any $\mathscr{L}_A$ formula $\varphi(x)$, if $T \vdash \exists x\varphi(x)$, then $T \nvdash \neg\varphi(\bar{n})$ for some $n \in \mathbb{N}$, i.e., for any $\mathscr{L}_A$ formula $\varphi(x)$, if $T \vdash \neg\varphi(\bar{n})$ for all $n \in \mathbb{N}$, then $T \nvdash \exists x\varphi(x)$.

Theorem 1.50 (Gödel's First Incompleteness, the Original Version, Gödel)

Let $T \supseteq \mathsf{Q}$ be a recursively axiomatizable theory. If T is ω-consistent, then there is a sentence σ independent of σ such that $T \nvdash \sigma$ and $T \nvdash \neg\sigma$.

Proof.

Suppose that the predicate Be is represented by $\mathsf{be}(x, y)$ in $T \supseteq \mathsf{Q}$, and let $\mathsf{beb}(y) = \exists x \mathsf{be}(x, y)$, then it's easy to check that Beb is represented by $\mathsf{beb}(y)$. Furthermore let σ be the fixed point of $\neg\mathsf{beb}(y)$. Then

$$T \vdash \sigma \leftrightarrow \neg\mathsf{beb}(\ulcorner \sigma \urcorner).$$

It's suffices to show that σ is independent of T. ↻

Theorem 1.51 (Gödel's First Incompleteness, the Strengthened Version, Rosser)

Let $T \supseteq \mathsf{Q}$ be a recursively axiomatizable theory. If T is consistent, then there is a sentence σ independent of σ such that $T \nvdash \sigma$ and $T \nvdash \neg\sigma$.

Proof.

$$\mathsf{pro}(x) = \exists y[\mathsf{be}(y, x) \wedge \forall z \,\bar{<}\, y \neg \mathsf{be}(z, \bar{\neg}(x))],$$

where the recursive function $\sharp\alpha \to \sharp(\neg\alpha)$ is represented by the formula $\bar{\neg}(x)$, and if $x \equiv \ulcorner\alpha\urcorner$, then $\bar{\neg}(x) \equiv \ulcorner\neg\alpha\urcorner$. We can prove

$$T \vdash \alpha \Rightarrow T \vdash \mathsf{pro}(\ulcorner\alpha\urcorner) \;\;\&\;\; T \vdash \neg\alpha \Rightarrow T \vdash \neg\mathsf{pro}(\ulcorner\alpha\urcorner).$$

Let σ be the fixed point of $\neg\mathsf{pro}(x)$. Then

$$T \vdash \sigma \leftrightarrow \neg\mathsf{pro}(\ulcorner\sigma\urcorner). \tag{5.1}$$

It suffices to show that σ is independent of T. ↻

By-products

Lemma 1.52 (Non-representability)

Let $T \supseteq \mathsf{Q}$ be a recursively axiomatizable theory. If T is consistent, then $\sharp T$ is not representable in T.

Proof.

Suppose $\sharp T$ is represented by $\varphi(x)$. Then for any formula ϑ, $T \vdash \vartheta \Rightarrow T \vdash \varphi(\ulcorner\vartheta\urcorner)$ and $T \nvdash \vartheta \Rightarrow T \vdash \neg\varphi(\ulcorner\vartheta\urcorner)$. i.e.,

$$T \nvdash \vartheta \Leftrightarrow T \vdash \neg\varphi(\ulcorner\vartheta\urcorner). \tag{6.1}$$

Now let σ be the fixed point of $\neg\varphi(x)$, then

$$T \vdash \sigma \leftrightarrow \neg\varphi(\ulcorner\sigma\urcorner). \tag{6.2}$$

By (6.1) and (6.2) $T \vdash \sigma \Leftrightarrow T \nvdash \sigma$, a contradiction. ○

By-products

Theorem 1.53 (Tarski's Non-definability)

$\sharp\,\mathrm{Th}(\mathcal{N}) = \{\sharp\vartheta \mid \mathcal{N} \models \vartheta\}$ *is not definable in the standard arithmetic model* $\mathcal{N}$.

By-products

Corollary 1.54

$\mathrm{Th}(\mathcal{N})$ *is undecidable, i.e.,* $\sharp\,\mathrm{Th}(\mathcal{N})$ *is not recursive.*

Theorem 1.55 (Strong Undecidability of Q)

Let T *be a theory such that* $T \cup \mathsf{Q}$ *is consistent. Then* T *is undecidable.*

Corollary 1.56 (Church's Undecidability)

Fix the language $\mathscr{L}_A$*. Then the set of validities is undecidable, i.e.,* $\{\vartheta \in \mathscr{L}_A \mid\ \models \vartheta\}$ *is undecidable.*

Theorem 1.57

Hilbert's Tenth Problem *Is there an algorithm such that for any polynomial* $p(\overrightarrow{x})$ *with integer coefficients decides whether the equation* $p(\overrightarrow{x}) = 0$ *has a solution in* $\mathbb{Z}$*? The answer is* NO.

Outline

The General Proof Idea

- We first introduce three derivability conditions. Then clime over three mountains as well as in last section.
- If $\vdash_T \varphi$, then $\vdash_T \Box_T\varphi$, i.e., T satisfies D_1;
- $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T\varphi \to \Box_T\psi$, T satisfies D_2;
- $\vdash_T \Box_T\varphi \to \Box_T\Box_T\varphi$, i.e., T satisfies D_3;
- At last we will show $\vdash_T \mathsf{con}(T) \to \neg\Box_T\mathsf{con}(T)$ which follows $\nvdash_T \mathsf{con}(T)$.

The General Proof Idea

- We first introduce three derivability conditions. Then clime over three mountains as well as in last section.
- If $\vdash_T \varphi$, then $\vdash_T \Box_T\varphi$, i.e., T satisfies D_1;
- $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T\varphi \to \Box_T\psi$, T satisfies D_2;
- $\vdash_T \Box_T\varphi \to \Box_T\Box_T\varphi$, i.e., T satisfies D_3;
- At last we will show $\vdash_T \mathsf{con}(T) \to \neg\Box_T\mathsf{con}(T)$ which follows $\nvdash_T \mathsf{con}(T)$.

The General Proof Idea

- We first introduce three derivability conditions. Then clime over three mountains as well as in last section.
- If $\vdash_T \varphi$, then $\vdash_T \Box_T \varphi$, i.e., T satisfies D_1;
- $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T \varphi \to \Box_T \psi$, T satisfies D_2;
- $\vdash_T \Box_T \varphi \to \Box_T \Box_T \varphi$, i.e., T satisfies D_3;
- At last we will show $\vdash_T \mathsf{con}(T) \to \neg \Box_T \mathsf{con}(T)$ which follows $\nvdash_T \mathsf{con}(T)$.

The General Proof Idea

- We first introduce three derivability conditions. Then clime over three mountains as well as in last section.
- If $\vdash_T \varphi$, then $\vdash_T \Box_T\varphi$, i.e., T satisfies D_1;
- $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T\varphi \to \Box_T\psi$, T satisfies D_2;
- $\vdash_T \Box_T\varphi \to \Box_T\Box_T\varphi$, i.e., T satisfies D_3;
- At last we will show $\vdash_T \mathsf{con}(T) \to \neg\Box_T\mathsf{con}(T)$ which follows $\nvdash_T \mathsf{con}(T)$.

The General Proof Idea

- We first introduce three derivability conditions. Then clime over three mountains as well as in last section.
- If $\vdash_T \varphi$, then $\vdash_T \Box_T\varphi$, i.e., T satisfies D_1;
- $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T\varphi \to \Box_T\psi$, T satisfies D_2;
- $\vdash_T \Box_T\varphi \to \Box_T\Box_T\varphi$, i.e., T satisfies D_3;
- At last we will show $\vdash_T \mathsf{con}(T) \to \neg\Box_T\mathsf{con}(T)$ which follows $\nvdash_T \mathsf{con}(T)$.

Peano Arithmetic

Definition 2.1

Peano arithmetic is the theory $\mathsf{PA} = \Phi + P$, where $P = I \cup \{Q_1, Q_2, Q_4, Q_5, Q6, Q_7\}$ and

$$\begin{array}{rcl}
Q_1 & : & \forall x \bar{S} x \bar{\not\equiv} \bar{0}; \\
Q_2 & : & \forall x \forall y (\bar{S} x \bar{\equiv} \bar{S} y \to x \bar{\equiv} y); \\
Q_4 & : & \forall x (x \bar{+} \bar{0} \bar{\equiv} x); \\
Q_5 & : & \forall x \forall y (x \bar{+} \bar{S} y \bar{\equiv} \bar{S}(x \bar{+} y)); \\
Q_6 & : & \forall x (x \bar{\times} \bar{0} \bar{\equiv} x); \\
Q_7 & : & \forall x \forall y (x \bar{\times} \bar{S} y \bar{\equiv} x \bar{\times} y \bar{+} x) \\
I & : & [\varphi(\bar{0}) \wedge \forall y(\varphi(y) \to \varphi(\bar{S} y))] \to \forall x \varphi(x), \\
 & & \varphi(x) \text{ is an } \mathscr{L}_A \text{ formula with } \mathrm{Fr}(\varphi) = \{x\} \text{ and } y \notin \mathrm{Vr}(\varphi).
\end{array}$$

We call I the set of induction axioms; given any set Θ of formulas with x free only of $\mathscr{L}_A$, then

$$I\Theta = \{[\varphi(\bar{0}) \wedge \forall y(\varphi(y) \to \varphi(\bar{S} y))] \to \forall x \varphi(x) | \varphi(x) \in \Theta, \mathrm{Fr}(\varphi) = \{x\} \text{ and } y \notin \mathrm{Vr}(\varphi)\}.$$

Peano Arithmetic

Lemma 2.2

$\mathsf{PA} = \mathsf{Q} + I$, *and so* PA *is the extension of* Q *generated by induction axioms.*

Proof.

Since $Q_3 : \forall x(x \not\equiv \bar{0} \rightarrow \exists y(x \equiv \bar{S}y))$, then consider the induction axiom $[\varphi(\bar{0}) \wedge \forall y(\varphi(y) \rightarrow \varphi(\bar{S}y))] \rightarrow \forall x\varphi(x)$, where

$$\varphi(x) = x \not\equiv \bar{0} \rightarrow \exists y(x \equiv \bar{S}y).$$

$\circlearrowleft$

Theorem 2.3 (Σ_1-completeness of PA)

For any Σ_1-sentence φ for $\mathcal{L}_A$, we have $\mathcal{N} \models \varphi$ iff $\mathsf{PA} \vdash \varphi$.

Peano Arithmetic

PA can prove basic properties about $\bar{S}$, $\bar{+}$, $\bar{\times}$, $\bar{\leq}$, $\bar{<}$; and further prove:

Lemma 2.4 (Strong Induction Principle)

$\mathsf{PA} \vdash \forall z[(\forall y \mathrel{\bar{<}} z\varphi(y)) \to \varphi(z)] \to \forall x\varphi(x)$, *where* $\varphi(x)$ *is an* $\mathcal{L}_A$ *formula with* $\mathrm{Fr}(\varphi) = \{x\}$ *and* $y, z \notin \mathrm{Vr}(\varphi)$.

Lemma 2.5 (The Least Number Principle)

$\mathsf{PA} \vdash \exists x\varphi(x) \to \exists x[\varphi(x) \wedge \forall y \mathrel{\bar{<}} x \neg\varphi(x)]$, *where* $\varphi(x)$ *is an* $\mathcal{L}_A$ *formula with* $\mathrm{Fr}(\varphi) = \{x\}$ *and* $y \notin \mathrm{Vr}(\varphi)$.

Derivability Conditions

Notation 2.6

Let T be any recursively axiomatizable theory and φ be any formula for $\mathscr{L}_A$. Convent

$$\begin{array}{rcccl} \Box_T(y) & = & \mathsf{beb}_T(y) & = & \exists x \mathsf{be}_T(x, y), \\ \Box_T\varphi & = & \Box_T(y; \ulcorner \varphi \urcorner) & = & \Box_T(\ulcorner \varphi \urcorner). \end{array}$$

Note that, $\Box_T(y)$ is a formula with a free variable y, while $\Box_T\varphi$ is a sentence no matter whether φ has free variables.

Derivability Conditions

Definition 2.7

Let T be any recursively axiomatizable theory and φ, ψ be any $\mathscr{L}_A$-sentences. The three derivability conditions are

$$
\begin{aligned}
D_1 &\ :\ \text{if } \vdash_T \varphi, \text{ then } \vdash_T \Box_T \varphi; \\
D_2 &\ :\ \vdash_T \Box_T(\varphi \to \psi) \to \Box_T \varphi \to \Box_T \psi; \\
D_3 &\ :\ \vdash_T \Box_T \varphi \to \Box_T \Box_T \varphi.
\end{aligned}
$$

Derivability Conditions

Lemma 2.8

Suppose T satisfy D_1 and D_2, then it also satisfies

$$D_0 \quad : \quad \textit{if } \varphi \vdash_T \psi, \textit{ then } \Box_T\varphi \vdash_T \Box_T\psi.$$

Corollary 2.9

If $\vdash_T \varphi \leftrightarrow \psi$, then $\vdash_T \Box_T\varphi \leftrightarrow \Box_T\psi$.

Definition 2.10

$\mathsf{con}(T) = \neg\Box_T \bar{0} \overline{\neq} \bar{0} = \neg\mathsf{beb}_T(\ulcorner \bar{0} \overline{\neq} \bar{0} \urcorner)$.

Corollary 2.9 tells us that in Definition 2.10 $\bar{0} \overline{\neq} \bar{0}$ could be replaced by any sentence equivalent to $\bot$, and so we may also set $\mathsf{con}(T) = \neg\Box_T\bot$.

T Satisfies D_1

Lemma 2.11

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{Q}$. Then T satisfies D_1, i.e., if $\vdash_T \varphi$, then $\vdash_T \Box_T\varphi$.

Proof.

Assume $\vdash_T \varphi$ and let n be the code of φ. Since the predicate Be is recursive, then by the Representability theorem we have $\vdash_T \mathsf{be}_T(\bar{n}, \ulcorner\varphi\urcorner)$, and so $\vdash_T \exists x \mathsf{be}_T(x, \ulcorner\varphi\urcorner)$, i.e., $\vdash_T \Box_T\varphi$. ↺

Definition 2.12

We say a recursive function $f : \mathbb{N}^k \to \mathbb{N}$ is provably recursive, or Σ_1-definable in $T \supseteq \mathsf{PA}$ if, there is a Σ_1 formula $\delta_f(\overrightarrow{x}, y)$ such that

$$\begin{array}{ll} T \vdash & \delta_f(\overrightarrow{n}, \overline{f(\overrightarrow{n})}) \qquad \text{for any } n_0, \cdots, n_{k-1} \in \mathbb{N}, \\ T \vdash & \forall \overrightarrow{x} \exists! y \delta_f(\overrightarrow{x}, y). \end{array}$$

We say a recursive predicate $P \subseteq \mathbb{N}^k$ is provably recursive, or Σ_1-definable in $T \supseteq \mathsf{PA}$ if, there is some Σ_1 formula $\delta_P(\overrightarrow{x})$ for $\mathscr{L}_A$ such that for any $n_0, \cdots, n_{k-1} \in \mathbb{N}$

$$P(\overrightarrow{n}) \Leftrightarrow T \vdash \delta_P(\overrightarrow{x}).$$

Lemma 2.13

In Definition 2.12, $\delta_f(\overrightarrow{x})$ and $\delta_P(\overrightarrow{x})$ are T-definitions for f and P respectively.

Lemma 2.14

The following are probably recursive in PA.

1. *the division relation* $d|x$*;*
2. *the reminder function* $\mathrm{rem}(x, d) = r$*;*
3. *"p is a prime"* $\mathrm{prime}(p)$*;*
4. *the binary maximum function* $\max(m, n)$*;*
5. *the coprime relation* $\mathrm{coprime}(m, n)$.

Proof.

(1) $\exists q < x(q \times d = x)$ (here we assuming that $0 \mid n$ iff $n = 0$);
(2) $[r < d \land \exists q < x(x = q \times d + r)] \lor (d = 0 \land r = 0)$;
(3) $p \neq 1 \land \forall d < p(d \mid p \to (d = 1 \lor d = p))$;
(4) $(m \leq n \land z = m) \lor (n < m \land z = n)$;
(5) $\forall d < \max(m, n)(d \mid m \land d \mid n \to d = 1)$.

- For a function, for example, $\max(m,n)$, we formalized it as a term function $\mathsf{max}(m,n)$. In other words, for any m,n the value $\mathsf{max}(m,n) = \overline{\max(m,n)}$ is a term. And in fact strictly speaking

$$(m \leq n \wedge z = m) \vee (n < m \wedge z = n)$$

should be written as

$$(\,\overline{m} \mathrel{\overline{\leq}} \overline{n} \,\wedge z \mathrel{\overline{=}} \overline{m}\,) \vee (\,\overline{n} \mathrel{\overline{<}} \overline{m} \,\wedge z \mathrel{\overline{=}} \overline{n}\,).$$

- For a function, for example, $\max(m, n)$, we formalized it as a term function $\mathsf{max}(m, n)$. In other words, for any m, n the value $\mathsf{max}(m, n) = \overline{\max(m, n)}$ is a term. And in fact strictly speaking

$$(m \leq n \wedge z = m) \vee (n < m \wedge z = n)$$

should be written as

$$(\,\overline{m} \mathrel{\overline{\leq}} \overline{n} \wedge z \mathrel{\overline{=}} \overline{m}\,) \vee (\,\overline{n} \mathrel{\overline{<}} \overline{m} \wedge z \mathrel{\overline{=}} \overline{n}\,).$$

- For a predicate, for example, prime(p), we formalized it as a formula function $\mathsf{prime}(p)$. In other words, for any p the value $\mathsf{prime}(p)$ is a formula.

- For a function, for example, $\max(m, n)$, we formalized it as a term function $\mathsf{max}(m, n)$. In other words, for any m, n the value $\mathsf{max}(m, n) = \overline{\max(m, n)}$ is a term. And in fact strictly speaking

$$(m \leq n \wedge z = m) \vee (n < m \wedge z = n)$$

should be written as

$$(\overline{m} \mathrel{\overline{\leq}} \bar{n} \wedge z \mathrel{\overline{=}} \overline{m}) \vee (\bar{n} \mathrel{\overline{<}} \overline{m} \wedge z \mathrel{\overline{=}} \bar{n}).$$

- For a predicate, for example, $\mathrm{prime}(p)$, we formalized it as a formula function $\mathsf{prime}(p)$. In other words, for any p the value $\mathsf{prime}(p)$ is a formula.
- And we note that there is some harmoniousness in the some defined formulas. For example,

$$\forall d < \mathsf{max}(m, n)(d = 9 \vee d = 10).$$

Strictly speaking it's should written as

$$\forall d \mathrel{\overline{<}} \mathsf{max}(m, n)(d \mathrel{\overline{=}} \bar{9} \vee d \mathrel{\overline{=}} \overline{10}).$$

The reason why we still write in the former form is to emphasize that $\mathsf{max}(m, n)$ has been formalized.

We should formulate some theorems such as Euclid lemma, Chinese reminder theorem and Gödel's β-function lemma in PA to formulate finite sequences.

Definition 2.15

Let finseq(s) be the formula

$$\exists c, k < s[s = \pi(c,k) \wedge \forall m < s(m < c \to \exists i < k(\beta(m,i) \neq \beta(c,i)))].$$

And set length(s) $= \pi_2(s)$ and value(s, i) $= \beta(\pi_1(s), i)$.

$$\begin{aligned}
\pi(x,y) &= \frac{1}{2}(x+y)(x+y+1)+x; \\
\pi_1(z) &= \mu x[\exists y \leq z(\pi(x,y) = z)]; \\
\pi_2(z) &= \mu y[\exists x \leq z(\pi(x,y) = z)].
\end{aligned}$$

Now we can formulate connection operation in PA. Consider the formula $\varphi(u, v, s)$

$$\begin{gathered}\mathsf{finseq}(s) \wedge \mathsf{length}(s) = \mathsf{length}(u) + \mathsf{length}(v) \\ \wedge[\forall i < \mathsf{length}(u) - 1\ (s)_i = u_i] \wedge [\forall i < \mathsf{length}(v) - 1\ (s)_{\mathsf{length}(u)+i} = v_i].\end{gathered}$$

Lemma 2.16

PA $\vdash \forall u \forall v \exists ! s \varphi(u, v, s)$. *So $\varphi(u, v, s)$ defines a provably recursive function in* PA*, and it is represented as $u \hat{}\, v$ and called formalized connection operation.*

Definition 2.17

Assign to every symbol of $\mathscr{L}_A$ a number term.

ζ	$\forall$	$\bar{0}$	$\bar{S}$	$\bar{+}$	$\bar{\times}$	$\bar{(}$	$\bar{)}$	$\bar{\neg}$	$\bar{\rightarrow}$	$\bar{=}$	x_0	x_1	x_2	$\cdots$
$\ulcorner\zeta\urcorner$	$\bar{1}$	$\bar{3}$	$\bar{5}$	$\bar{7}$	$\bar{9}$	$\overline{11}$	$\overline{13}$	$\overline{15}$	$\overline{17}$	$\overline{19}$	$\overline{21}$	$\overline{23}$	$\overline{25}$	$\cdots$

Lemma 2.18

The predicate variable(x) *is provably recursive in* PA.

Proof.

Consider the Σ_1 formula variable$(x) : \exists y < x(x = 2 \times y + 21)$. ⟳

Lemma 2.19

The predicate term(t) *is provably recursive in* PA.

Proof.

It's defined by the formula:

$$\exists s[\mathsf{finseq}(s) \wedge 0 < \mathsf{length}(s) \wedge s_{\mathsf{length}(s)-1} = t \wedge \forall i < \mathsf{length}(s) - 1 (s_i = \ulcorner 0 \urcorner \vee \varphi(s,i) \vee \psi(s,i))],$$

where

$$\varphi(s,i): \ \exists x < s_i(\mathsf{variable}(x) \wedge s_i = \langle x \rangle)$$

and

$$\psi(s,i): \ \exists m,n < i(s_i = \langle \ulcorner S \urcorner \rangle \hat{\ } s_m \vee s_i = \langle \ulcorner + \urcorner \rangle \hat{\ } s_m \hat{\ } s_n \vee s_i = \langle \ulcorner \times \urcorner \rangle \hat{\ } s_m \hat{\ } s_n).$$

And clearly it's Σ_1.

Also similarly we

- use the formula $\mathsf{formula}(x)$ to define the predicate formula(x) which is a formalization of "formulas";
- use the formula $\chi_\neg(x,y):\ x = \langle\ulcorner(\urcorner\rangle^\frown\langle\ulcorner\neg\urcorner\rangle^\frown y^\frown\langle\ulcorner)\urcorner\rangle$ to define the function $\dot\neg(x)$ which is a formalization of $\neg$;
- use the formula $\chi_\to(x,y,z):\ x = \langle\ulcorner(\urcorner\rangle^\frown y^\frown\langle\ulcorner\to\urcorner\rangle^\frown z^\frown\langle\ulcorner)\urcorner\rangle$ to define the function $\dot\to(x,y)$ which is a formalization of $\to$;
- use the formula $\mathsf{axiom}_T(x)$ to define the predicate axiom$_T(x)$ which is a formalization of some recursively axiomatizable theory T;
- use the formula $\mathsf{modpen}(x,y,z)\ :\ \chi_\to(x,y,z)\wedge\mathsf{formula}(y)\wedge\mathsf{formula}(z)$ to define the predicate modpen(x,y,z) which is a formalization of modus ponens rule.

Theorem 2.20 (Formalized Provability)

Both the binary predicate $\mathrm{Be}_T(x, y)$ *and unary predicate* $\mathrm{Beb}_T(y)$ *are provably recursive in* $T \supseteq \mathsf{PA}$.

Proof.

$\mathrm{Be}_T(x, y)$ is defined by the Σ_1 formula $\mathsf{Be}_T(x, y)$

$$\begin{aligned} & \mathsf{finseq}(x) \ \wedge \ s_{\mathsf{length}(x)-1} = y \\ \wedge \quad & \forall i < \mathsf{length}(x) - 1[\mathsf{axiom}_T(x_i) \vee \exists m, n < i\mathsf{modpen}(x_m, x_n, x_i)]. \end{aligned}$$

So $\mathrm{Beb}_T(y)$ is defined by the Σ_1 formula $\mathsf{Beb}_T(y) = \exists x \mathsf{Be}_T(x, y)$. ⭮

T Satisfies D_2

Lemma 2.21

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. Then T satisfies D_2, i.e., $\vdash_T \Box_T(\varphi \to \psi) \to \Box_T\varphi \to \Box_T\psi$.

Proof.

Suppose u and v satisfies $\mathsf{Be}_T(u, \ulcorner\varphi \to \psi\urcorner)$ and $\mathsf{Be}_T(v, \ulcorner\varphi\urcorner)$ respectively. It suffices to show

$$T \vdash \mathsf{Be}_T(u, \ulcorner\varphi \to \psi\urcorner) \to \mathsf{Be}_T(v, \ulcorner\varphi\urcorner) \to \mathsf{Be}_T(u\hat{\ }v\hat{\ }\langle\ulcorner\psi\urcorner\rangle, \ulcorner\psi\urcorner).$$

Set $s = u\hat{\ }v\hat{\ }\langle\ulcorner\psi\urcorner\rangle$. It's easy to show:

- $T \vdash \mathsf{finseq}(s)$;
- $T \vdash s_{\mathsf{length}(s)-1} = \ulcorner\psi\urcorner$;
- $T \vdash \forall i < \mathsf{length}(s) - 1[\mathsf{axiom}_T(s_i) \vee \exists m, n < i\, \mathsf{modpen}(s_m, s_n, s_i)]$.

Then by the definition, we have $\mathsf{Be}_T(u\hat{\ }v\hat{\ }\langle\ulcorner\psi\urcorner\rangle, \ulcorner\psi\urcorner)$. ↻

T Satisfies D_3

$$D_3 : \vdash_T \Box_T \varphi \to \Box_T \Box_T \varphi.$$

If $T \vdash \varphi$ then $T \vdash \mathsf{beb}_T(\ulcorner \varphi \urcorner)$.

Σ_1-completeness: $\vdash_T \varphi(\overrightarrow{x}) \to \Box_T \lfloor \varphi(\overrightarrow{x}) \rfloor$ for any Σ_1 formula $\varphi(\overrightarrow{x})$.

Recall the recursive function $f(n) = \sharp\,\bar{n} = \sharp\,\bar{S}^n\bar{0} = \mathsf{tnum}(n)$. Consider the formula $\varphi(x,y)$

$$\exists s[\mathsf{finseq}(s)\wedge\mathsf{length}(s)\bar{=}x\bar{+}\bar{1}\wedge s_0\bar{=}\ulcorner 0\urcorner\wedge s_{x+1}\bar{=}y\wedge(\forall i < x s_{i+1}\bar{=}\langle\ulcorner S\urcorner\rangle\hat{\ } s_i)].$$

Lemma 2.22

$\mathsf{PA}\vdash\forall x\exists! y\varphi(x,y)$. *So $f(n)$ defined by φ is provably recursive in* PA, *and the corresponding formalized function is* $\mathsf{tnum}(x) = \overline{\mathsf{tnum}}(x)$.

For any n, $\mathsf{tnum}(n)$ is the term $\overline{\ulcorner\bar{n}\urcorner} = \bar{S}^{\ulcorner\bar{n}\urcorner}\bar{0} = \bar{S}^{\ulcorner\bar{S}^n\bar{0}\urcorner}\bar{0}$.

Lemma 2.23

The function $\mathrm{fvariable}(x) = y = 2x + 21$ *is provably recursive in* PA. *And corresponding formalized function is* $\mathsf{fvariable}(x) = \overline{\mathrm{fvariable}}(x)$.

Lemma 2.24

The function $\mathrm{sub}(\mathrm{tnum}(x), \mathrm{fvariable}(y), z)$ *is provably recursive in* PA. *And the formalized function is*

$$\mathsf{sub}(\mathsf{tnum}(x), \mathsf{fvariable}(y), z) = \overline{\mathrm{sub}(\mathrm{tnum}(x), \mathrm{fvariable}(y), z)}\,.$$

Note that the values of $\mathsf{sub}(\mathsf{tnum}(x), \mathsf{fvariable}(y), z)$ *are terms.*

We illustrate how $\mathrm{sub}(\mathrm{tnum}(x), \mathrm{fvariable}(y), z)$ operates the results by setting $x = 3, y = 4$ and $z = \ulcorner x_4 \equiv x_6 \urcorner$:

- decode z as a formula $x_4 \equiv x_1$;
- find all the free variables which are signed on 4, i.e., all the free x_4; replace all the free x_4 by x_3;
- get $x_3 \equiv x_6$; set $\mathrm{sub}(\mathrm{tnum}(x), \mathrm{fvariable}(y), z) = \ulcorner x_3 \equiv x_6 \urcorner$.

Since the whole process occurs "in" PA, then

$$\mathsf{PA} \vdash [\mathsf{sub}(\mathsf{tnum}(3), \mathsf{fvariable}(4), \ulcorner x_4 \equiv x_6 \urcorner)] \equiv [\,\bar{S}^{\ulcorner x_3 \equiv x_6 \urcorner}\bar{0}\,].$$

Compare the two x_4 with each other in

$$\mathrm{sub}(\mathrm{tnum}(x_4), \mathrm{fvariable}(4), \ulcorner x_4 \,\overline{\overline{=}}\, x_6 \urcorner).$$

It's not hard to see that

$$\begin{aligned} & \mathrm{sub}(\mathrm{tnum}(x_4), \mathrm{fvariable}(4), \ulcorner x_4 \,\overline{\overline{=}}\, x_6 \urcorner) \\ = \quad & \ulcorner x_{(\mathrm{tnum}(x_4)-21)/2} \,\overline{\overline{=}}\, x_6 \urcorner = \langle 19, \mathrm{tnum}(x_4), 33 \rangle \\ = \quad & \ulcorner \mathsf{fvariable}(x_4) \,\overline{\overline{=}}\, x_6 \urcorner. \end{aligned}$$

Clearly the first x_4 is free, while the second one is always "dead". Assign any value a (maybe not a standard element) to x_4, we would get a corresponding $\ulcorner \mathsf{fvariable}(a) \,\overline{\overline{=}}\, x_6 \urcorner$ which shows x_4 is free.

For convenience, we set $\mathsf{su}(x, y, z) = \mathsf{sub}(\mathsf{tnum}(x), \mathsf{fvariable}(y), z)$.

Definition 2.25

Suppose φ is an $\mathscr{L}_A$-formula such that $\mathrm{Fr}(\varphi) = \{x_{k_0}, \cdots, x_{k_{n-1}}\}$, and we may further assume $k_0 < \cdots < k_{n-1}$. Then

$$\Box_T\lfloor\varphi(\overrightarrow{x})\rfloor = \Box_T \mathsf{su}(x_{k_{n-1}}, k_{n-1}, \cdots, \mathsf{su}(x_{k_1}, k_1, \mathsf{su}(x_{k_0}, k_0, \ulcorner\varphi\urcorner))\cdots).$$

$$\Box_T\lfloor\varphi(x)\rfloor = \Box_T\mathsf{su}(x,k,\ulcorner\varphi\urcorner) = \Box_T\mathsf{sub}(\mathsf{tnum}(x),\mathsf{fvariable}(k),\ulcorner\varphi\urcorner).$$

- Clearly $\Box_T\lfloor\varphi\rfloor = \Box_T\ulcorner\varphi\urcorner$ if φ is a sentence;
- $\Box_T\lfloor\varphi(\overrightarrow{x})\rfloor$ and $\varphi(\overrightarrow{x})$ have the same free variables, while $\Box_T\ulcorner\varphi(\overrightarrow{x})\urcorner$ has no variables;
- Sometimes considering of readability we write some common variables x, y, z instead of x_{k_n} since we are very clear that which variable should be refried;
- It's obvious that $\vdash_T \Box_T\lfloor\varphi(\overrightarrow{x})\rfloor$ and $\vdash_T \Box_T\ulcorner\varphi(\overrightarrow{x})\urcorner$ are different.

Lemma 2.26 (Formalized D_1)

For any $\mathscr{L}_A$-formula φ, if $\vdash_T \varphi$, then $\vdash \Box_T \lfloor \varphi \rfloor$.

Lemma 2.27 (Formalized D_2)

For any $\mathscr{L}_A$-formulas φ and ψ, $\vdash_T \Box_T \lfloor \varphi \to \psi \rfloor \to \Box_T \lfloor \varphi \rfloor \to \Box_T \lfloor \psi \rfloor$.

Lemma 2.28

Suppose $\varphi(x_0)$ is a formula with only x free (the general case is similar), and x_k is free for x_1 in φ, then

1. $\vdash_T \Box_T\lfloor\varphi(x_0;\bar{0})\rfloor \leftrightarrow (\Box_T\lfloor\varphi\rfloor)(x_0;\bar{0})$;
2. $\vdash_T \Box_T\lfloor\varphi(x_0;x_k)\rfloor \leftrightarrow (\Box_T\lfloor\varphi\rfloor)(x_0;x_k)$;
3. $\vdash_T \Box_T\lfloor\varphi(x_0;\bar{S}x_k)\rfloor \leftrightarrow (\Box_T\lfloor\varphi\rfloor)(x_0;\bar{S}x_k)$.

Theorem 2.29 (Formalized Σ_1-completeness)

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. Then $\vdash_T \varphi \to \Box_T\lfloor\varphi\rfloor$ for any Σ_1 formula.

T Satisfies D_3

Lemma 2.30

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. Then T satisfies D_3, i.e., $\vdash_T \Box_T\varphi \to \Box_T\Box_T\varphi$.

Proof.

This follows from formalized Σ_1-completeness since $\Box_T\varphi$ is Σ_1 and $\Box_T\lfloor\Box_T\varphi\rfloor = \Box_T\Box_T\varphi$ for which $\Box_T\varphi$ is a sentence. ∎

Theorem 2.31 (Formalized Gödel's Second Incompleteness, FGSIT)

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. If T is consistent, then $\vdash_T \mathsf{con}(T) \to \neg\Box_T\mathsf{con}(T)$.

Proof.

By fixed point lemma 1.48, for $\neg\Box_T(y)$, there is some σ such that

$$\vdash_T \sigma \leftrightarrow \neg\Box_T\sigma. \tag{6.1}$$

We claim that

$$\vdash_T \sigma \leftrightarrow \mathsf{con}(T). \tag{6.2}$$

By (6.2) and D_0 we have

$$\vdash_T \Box_T\sigma \leftrightarrow \Box_T\mathsf{con}(T). \tag{6.3}$$

And then by (6.1), (6.2) and (6.3) we have

$$\vdash_T \mathsf{con}(T) \leftrightarrow \neg\Box_T\mathsf{con}(T) \tag{6.4}$$

as desired. ↺

Corollary 2.32 (Gödel's Second Incompleteness Theorem, GSIT)

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. If T is consistent, then $\nvdash_T \mathsf{con}(T)$.

Proof.

Suppose for sake of a contradiction that $\vdash_T \mathsf{con}(T)$. Then by D_1 we have $\vdash_T \Box_T \mathsf{con}(T)$, and then by (6.4) we have $\vdash_T \neg\mathsf{con}(T)$, a contradiction to the consistency of T. ↺

Incompleteness

There are three kinds of "completeness" for a theory T in this material:

- syntactical completeness: T is (sytactically) complete if for any formula φ either $T \vdash \varphi$ or $T \vdash \neg\varphi$;
- meta-semantical completeness: T is meta-semantically complete if T can prove any property related to T which is out of T;
- semantical completeness: T is complete if, for any φ we have $T \vdash \varphi$ if $T \models \varphi$.

Clearly Gödel's completeness theorem tells us that T has semantical completeness. So, there are two kinds of "incompleteness" corresponding to the first two kinds of "completeness". And Gödel's first and second incompleteness theorems tells us that a theory satisfying some conditions owns neither of the first two "completeness" as above.

Let's give a definition about the Goodstein sequences which is not so rigid:

- given natural numbers $m \geq 1$ and $n \geq 2$, we can define base n representation of m and pure base n representation of m. We just use one example to illustrate the concept: say $m = 13$ and $n = 2$, $13 = 2^3 + 2^2 + 1$ (which is base 2 representation)$= 2^{2+1} + 2^2 + 1$ (which is pure base 2 representation).
- we define the Goodstein sequence $\langle g_n | n \in \mathbb{N} \rangle$ beginning from m by recursion:
  - $g_0 = m$;
  - Given g_n, we get g_{n+1} as follows: write g_n in pure $n+2$ representation, replacing each base $n+2$ by $n+3$, and then subtract 1.

For example, the Goodstein sequence beginning from $m = 13$ runs as follows:

$$
\begin{array}{ccccclll}
g_0 & = & 13 & = & 2^{2+1} + 2^2 + 1 & 2 \rightarrowtail 3 & 3^{3+1} + 3^3 + 1 = 109 \\
g_1 & = & 108 & = & 3^{3+1} + 3^3 & 3 \rightarrowtail 4 & 4^{4+1} + 4^4 = 1280 \\
g_2 & = & 1279 & = & 4^{4+1} + 3 \cdot 4^3 + 3 \cdot 4^2 + 3 \cdot 4 + 3 & 4 \rightarrowtail 5 & 5^{5+1} + 3 \cdot 5^3 + 3 \cdot 5^2 + 3 \cdot 5 + 3 = 16093 \\
g_3 & = & 16092 & = & 5^{5+1} + 3 \cdot 5^3 + 3 \cdot 5^2 + 3 \cdot 5 + 2 & 5 \rightarrowtail 6 & 6^{6+1} + 3 \cdot 6^3 + 3 \cdot 6^2 + 3 \cdot 6 + 2 = 280712 \\
g_4 & = & 280711 & = & 6^{6+1} + 3 \cdot 6^3 + 3 \cdot 6^2 + 3 \cdot 6 + 1 & 6 \rightarrowtail 7 & 7^{7+1} + 3 \cdot 7^3 + 3 \cdot 7^2 + 3 \cdot 7 + 1 = 5765999 \\
g_5 & = & 5765998 & = & 7^{7+1} + 3 \cdot 7^3 + 3 \cdot 7^2 + 3 \cdot 7 & 7 \rightarrowtail 8 & 8^{8+1} + 3 \cdot 8^3 + 3 \cdot 8^2 + 3 \cdot 8 = 134219480 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Theorem 2.33 (Goodstien)

Every Goodstein sequence ends in 0.

(Sketch).

Replacing each base by ω in each term of g_n, we will get a descending sequence $\langle \alpha_n | n \in \mathbb{N} \rangle$ of ordinals. By Regularity of Axiom, the descending ordinal sequences must be end in 0, and so does the Goodstein sequence. We still use an example to illustrate the proof idea:

$$
\begin{array}{llllllll}
g_0 & = & 13 & = & 2^{2+1} + 2^2 + 1 & 2 \rightarrowtail \omega & \omega^{\omega+1} + \omega^{\omega} + 1 \\
g_1 & = & 108 & = & 3^{3+1} + 3^3 & 3 \rightarrowtail \omega & \omega^{\omega+1} + \omega^{\omega} \\
g_2 & = & 1279 & = & 4^{4+1} + 3 \cdot 4^3 + 3 \cdot 4^2 + 3 \cdot 4 + 3 & 4 \rightarrowtail \omega & \omega^{\omega+1} + 3 \cdot \omega^3 + 3 \cdot \omega^2 + 3 \cdot \omega + 3 \\
g_3 & = & 16092 & = & 5^{5+1} + 3 \cdot 5^3 + 3 \cdot 5^2 + 3 \cdot 5 + 2 & 5 \rightarrowtail \omega & \omega^{\omega+1} + 3 \cdot \omega^3 + 3 \cdot \omega^2 + 3 \cdot \omega + 2 \\
g_4 & = & 280711 & = & 6^{6+1} + 3 \cdot 6^3 + 3 \cdot 6^2 + 3 \cdot 6 + 1 & 6 \rightarrowtail \omega & \omega^{\omega+1} + 3 \cdot \omega^3 + 3 \cdot \omega^2 + 3 \cdot \omega + 1 \\
g_5 & = & 5765998 & = & 7^{7+1} + 3 \cdot 7^3 + 3 \cdot 7^2 + 3 \cdot 7 & 7 \rightarrowtail \omega & \omega^{\omega+1} + 3 \cdot \omega^3 + 3 \cdot \omega^2 + 3 \cdot \omega \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

Theorem 2.34 (Löb)

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$.

1. $\vdash_T \Box_T(\Box_T\varphi \to \varphi) \to \Box_T\varphi$;
2. *If* $\vdash \Box_T\varphi \to \varphi$, *then* $\vdash_T \varphi$.

Corollary 2.35

Suppose T is a recursively axiomatizable theory with $T \supseteq \mathsf{PA}$. Then $\top$ is the only fixed point of $\Box_T(y)$ up to the logical equivalence in T.

- We also call $\vdash_T \Box_T(\Box_T\varphi \to \varphi) \to \Box_T\varphi$ as D_4 which could be also regarded as a derivability condition;
- It's easy to see that in Theorem 2.34 D_4 is the formalization of (2) in T, and so we call "If $\vdash \Box_T\varphi \to \varphi$, then $\vdash_T \varphi$" as ${D_4}^{\diamond}$;
- Since $\vdash_T \mathsf{con}(T)$, i.e., $\vdash_T \Box_T\bot \to \bot$, implies $\vdash_T \bot$ by ${D_4}^{\diamond}$, then $D_4 \Rightarrow {D_4}^{\diamond} \Rightarrow$GSIT;
- Since D_4 implies FGSIT for $\varphi = \bot$ by contraposition, then $D_4 \Rightarrow$FGSIT$\Rightarrow$GSIT;
- So Löb theorem is stronger than Gödel's second incompleteness theorem which is not obvious at first glance.

Corollary 2.36

Suppose T is a recursively axiomatizable and consistent theory with $T \supseteq \mathsf{PA}$ and $T^\diamond = T + \neg\mathsf{con}(T)$. Then $T^\diamond$ is consistent and $T^\diamond \vdash \neg\mathsf{con}(T^\diamond)$ and $T^\diamond$ is ω-inconsistent.

- $\mathsf{PA}^\diamond$ is inconsistent in $\mathsf{PA}^\diamond$ itself although $\mathsf{PA}^\diamond$ is consistent out of $\mathsf{PA}^\diamond$;
- The consistent $\mathsf{PA}^\diamond$ can prove its inconsistency but never prove its consistency;
- $\mathsf{PA}^\diamond$ is a typical ω-inconsistent theory;
- There is some consistent theory T such that $T + \mathsf{con}(T)$ is inconsistent ($T = \mathsf{PA}^\diamond$).

By-products Meta-theoretic Properties of T

Many meta-theoretic properties of T could be formalized in T using provability operator $\Box_T$ and sentence schemata as above:

$\neg\mathsf{con}(T)$	:	$\Box_T \bot$	provable inconsistency,
secomp	:	$\varphi \to \Box_T \varphi$	semantical completeness,
sycomp	:	$\Box_T \varphi \vee \Box_T \neg\varphi$	syntactic completeness,
ω-comp	:	$\forall x \Box_T \lfloor \varphi(x) \rfloor \to \Box_T \forall x \varphi(x)$	ω-completeness.

Theorem 2.37

Suppose T is a recursively axiomatizable theory with $T \supseteq$ PA. Then the following sentences are logically equivalent in T:

1. $\neg\mathsf{con}(T)$;
2. $\varphi \to \Box_T \varphi$;
3. $\Box_T \varphi \vee \Box_T \neg\varphi$;
4. $\forall x \Box_T \lfloor \varphi(x) \rfloor \to \Box_T \forall x \varphi(x)$.

All the properties above hold for theories $T = T^{\diamond}$.

Outline

References

G. Boolos.
The Provability of Logic.
Cambridge University Press, 1st edition, 2003.

N. J. Cutland.
Computability: an Introduction to Recursive Function Theory.
Cambridge University Press, 1980.

P. G. Hinman.
Fundamentals of Mathematical Logic.
AK Peters Wellesley, 2005.

K. Hrback and T. Jech.
Introduction to Set Theory.
Marcel Dekker Inc., 3rd edition, 1999.

K. Kunen.
The Foundations of Mathematics.
Colleage Publications, 2009.

W. Rautenberg.
A Concise Introduction to Mathematical Logic.
Springer, 2006.